

# Multidimensional Image Processing Laboratory

## Manual of Hardware, Software, and Protocol

Krishnan Ramaswamy, Shu-Yen Wan, Allen Austin, James Helferty, William E. Higgins

Department of Electrical Engineering, CSE, and Bioengineering  
The Pennsylvania State University

December 5, 2000

# Contents

<b>1</b>	<b>Lab Organization</b>	<b>4</b>
1.1	Lab Manuals . . . . .	4
1.2	IP/Host Information . . . . .	5
1.3	PC Configuration . . . . .	5
1.3.1	paige.ee.psu.edu . . . . .	7
1.3.2	clemente.ee.psu.edu . . . . .	8
1.3.3	cyyoung.ee.psu.edu . . . . .	9
1.3.4	hornsby.ee.psu.edu . . . . .	10
1.3.5	koufax.ee.psu.edu . . . . .	11
1.3.6	williemays.ee.psu.edu . . . . .	12
1.3.7	musial.ee.psu.edu . . . . .	13
1.4	Other Hardware . . . . .	13
1.5	Sun Configuration . . . . .	13
1.5.1	aaron.ee.psu.edu . . . . .	14
1.5.2	doubleday.ee.psu.edu . . . . .	14
1.5.3	gehrig.ee.psu.edu . . . . .	15
1.5.4	yastrzemski.ee.psu.edu . . . . .	15
<b>2</b>	<b>Lab Administration</b>	<b>16</b>
2.1	PC File Systems and Security . . . . .	16
2.2	Image Databases . . . . .	16
2.2.1	PC Database . . . . .	16
2.2.2	Sun Database . . . . .	17
2.3	Lab Web Page . . . . .	17
2.3.1	Personal Home Page on Lab Web Page . . . . .	17
2.3.2	Home Page Maintenance . . . . .	18
2.4	PC Virus Scanning . . . . .	18
2.5	Lab ftp Area . . . . .	18
2.6	Sun File Management and Home Directory Quotas . . . . .	19
2.7	Miscellaneous Sun System/Lab details . . . . .	20
2.8	Procedures for Setting up New Sun Accounts . . . . .	20

<b>3</b>	<b>Equipment Maintenance and EE Staff</b>	<b>22</b>
3.1	PCs . . . . .	22
3.2	Sun Maintenance . . . . .	22
3.3	EE Department Staff . . . . .	23
<b>4</b>	<b>Miscellaneous Lab Items</b>	<b>24</b>
4.1	Shooting Photographs and Slides . . . . .	24
4.2	Powerpoint to 35mm Slides . . . . .	24
4.3	Phone Directory . . . . .	24
4.4	General BIBTEX area . . . . .	24
4.5	Lab Security . . . . .	25
4.6	Long-Distance Phone Calls . . . . .	25
<b>5</b>	<b>Code Specification</b>	<b>26</b>
5.1	Variable Names and Declarations . . . . .	26
5.2	Source Code Files (*.cpp and *.h) . . . . .	28
5.3	Classes, Object-Oriented Programming . . . . .	30
5.4	Source Modification and Maintenance . . . . .	32
<b>6</b>	<b>Available Software Packages</b>	<b>34</b>
6.1	Editing FILES, Creating DOCUMENTS, Drawing FIGURES . . . . .	34
6.1.1	PC . . . . .	34
6.1.2	UNIX . . . . .	35
6.2	IMAGE PROCESSING and DISPLAY . . . . .	36
6.2.1	PC . . . . .	36
6.2.2	UNIX . . . . .	37
6.3	PROGRAM DEVELOPMENT, COMPILATION, and DEBUGGING . . . . .	38
6.3.1	PC . . . . .	38
6.3.2	UNIX . . . . .	39
6.4	Miscellaneous packages . . . . .	39
6.4.1	PC . . . . .	39
6.4.2	UNIX . . . . .	40
6.5	Packages for EDITING FILES, CREATING FORMATTED DOCUMENTS, DRAW- ING FIGURES . . . . .	41
6.6	Packages for IMAGE PROCESSING AND DISPLAY . . . . .	44

6.7	Packages for PROGRAM DEVELOPMENT, COMPILATION, and DEBUGGING	49
6.8	Miscellaneous Packages . . . . .	53
<b>7</b>	<b>BACKUPS</b>	<b>56</b>
7.1	PC Back-up Plan . . . . .	56
7.1.1	Hardware and drivers . . . . .	56
7.1.2	Inserting and Removing Tapes . . . . .	56
7.1.3	Backup Software . . . . .	56
7.2	PC Tape Rotation Schedule . . . . .	57
7.3	Sun Back-up Plan . . . . .	58
7.4	Old Back-up Plan – superseded 6/1/98 . . . . .	60
7.5	Old Plan: ruth . . . . .	60
7.6	8mm Tape Back-up . . . . .	61
7.7	special . . . . .	61
7.8	Level 0 Dump on Solaris 2.X . . . . .	62

# 1 Lab Organization

## 1.1 Lab Manuals

Books and manuals can be found in the book cabinets in the lab. Separate boxes for each PC, such as paige, hornsby, etc., are here. A sample of books:

1. Windows 2000, NT
2. Visual C++ 6.0, 5.0; Visual Studio, Visual Basic, *et al.*
3. Rational Rose, UML
4. Microsoft Office, Adobe tools, Mathematica
5. Unix, Solaris, System administration (in closed cabinet)
6. Our internal software packages, such as VFX, Virtual Navigator, QUICKSEE, Contalgo, VIDA, ANALYZE, devguide, etc.
7. Sun documentation
8. printers, disk drives, and other peripherals.

*Please use the sign-up sheet in the lab to check manuals out. Keep the manuals for a day or so only.* Many manuals can be found on-line on the PCs and Suns.

*For the PCs*, these manuals are on `cyyoung:S\manual` and can be view with `ghostview` (or just click on the postscript manual).

*For the Suns*, you can follow the procedures below to view on-line help and manuals:

1. By using the  
`man` command  
while on a workstation, you can get "how-to" information for the specified command. Also, the tool "xman" is handy for this.
2. On-line manuals for some of the main image-analysis packages can be found in the area `/psu/manual`. These manuals can be read using the command  
`% xdvi <manual>` where "manual" is the name of the \*.dvi manual document for the specified package (either have the complete pathname or be in the directory with the \*.dvi file). A few of these manuals also exist as postscript (\*.ps) files – these can be read with `ghostview` or they can be printed out. A few of the manuals are also on-line in html form.

## 1.2 IP/Host Information

The available hostnames and IP addresses are as follows:

doubleday.ee.psu.edu	130.203.198.133	Sun sparc20
yastrzemski.ee.psu.edu	130.203.198.134	Sun sparc10 in WEH's office
gehrig.ee.psu.edu	130.203.198.135	Sun Ultra5
cobb.ee.psu.edu	130.203.198.137	SAME AS gehrig
hornsby.ee.psu.edu	130.203.198.138	Dell PC
aaron.ee.psu.edu	130.203.198.139	Sun Enterprise 3000, SAME AS ruth
no name (formerly ruth)	130.203.198.140	
cyyoung.ee.psu.edu	130.203.198.166	Dell PC
musial.ee.psu.edu	130.203.198.167	Lexmark network printer
jackie-r.ee.psu.edu	130.203.198.168	(WEH -- laptop)
clemente.ee.psu.edu	130.203.198.169	Dell workstation
paige.ee.psu.edu	130.203.198.236	Dell workstation
koufax.ee.psu.edu	130.203.198.238	Dell PC
williemays.ee.psu.edu	130.203.198.237	Dell PC
nolanryan.ee.psu.edu	130.203.199.185	(Tony S. -- laptop)
mathewson.ee.psu.edu	130.203.199.186	(Jim H. -- laptop)
spahn.ee.psu.edu	130.203.199.187	(Justin Yu -- laptop)

Unused AS OF 10/29/00:

ripken.ee.psu.edu	130.203.199.188
-------------------	-----------------

## 1.3 PC Configuration

The network configuration details for adding a new machine to our lab's PC network are the following:

Domain name for NT Server:	3DImage-S (paige is the PDC)
Network workgroup:	3DImage
Netmask:	255.255.255.0
Default Gateway	130.203.198.1
DNS:	130.203.197.3
	128.118.25.3
	130.203.1.4
WINS:	130.203.197.7
	130.203.197.4
DNS Suffixes:	ee.psu.edu
	psu.edu

The above information can be put into a PC configuration by (1) right-clicking on the Network Places, (2) select Local Area Network Connection, (3) hit Properties, (4) select Internet (TCP/IP) protocol, (5) hit Properties.

Please note that all PCs have OpenNT installed. Clemente is installed with additional OpenNTif and OpenNT SDK.

The configuration details for the individual PCs are given below.

### 1.3.1 paige.ee.psu.edu

130.203.198.236 primary domain controller (PDC)

#### Hardware:

Dell 6400 PII/MT Workstation 410  
Windows NT 4.0 Server, service pack 4  
3 years service from Dell  
dual 400 Mhz Pentium II  
100 Mhz bus  
21" monitor  
512 Mb ECC SDRAM, consisting of 4 128Mb DIMMs  
AGP video card: Guillemot Hercules 3D Prophet II GTS, 64MB memory  
12/24X SCSI CD ROM  
100 Mb SCSI Zip drive  
integrated 3Com Fast Etherlink XL 10/100 ethernet card  
floppy drive  
built-in Sound Blaster compatible audio  
mouse, keyboard

disk #1: 9Gb UW SCSI, 10K RPM, SCSI controller built in  
C: 2.0GB  
E: 3.0GB, applications  
Z: 2.0GB, user folders  
K: 1.6GB, EMPTY

disk #2: 9Gb UW SCSI, 10K RPM, SCSI controller built in  
D: 4.3GB, demos  
W: 4.3GB, microCT data, video(?)

#### Licensed Software:

Visual Studio 6.0 – from PSU EE  
Office 2000 – from PSU EE

#### Freeware:

EmTex  
Adobe Acrobat Reader  
GSView  
Winzip

#### Other:

service tag: FNFRM  
express service code: 262-877-62  
FAT to NTFS conversion done  
Bios v 2.01.05



### 1.3.2 clemente.ee.psu.edu

130.203.198.169      bronchoscopic testing environment

#### Hardware:

Dell 6400 PII/MT Workstation 410  
Windows NT 4.0, service pack 4  
3 years service from Dell  
dual 400 Mhz Pentium II  
100 Mhz bus  
21" monitor  
512 Mb ECC SDRAM, consisting of 4 128Mb DIMMs  
AGP video card: ATI Radeon 64MG DRAM  
12/24 Gb SCSI 4mm tape drive  
12/24X SCSI CD ROM  
100 Mb SCSI Zip drive  
integrated 3Com Fast Etherlink XL 10/100 ethernet card  
floppy drive  
built-in Sound Blaster compatible audio  
mouse, keyboard

disk #1:      9Gb UW SCSI, 10K RPM, SCSI controller built in  
C: 2.0GB

E: 3.0GB, freeware, license software  
P: 1.6GB, system virtual memory (1.0 GB)  
Z: 2.0GB, user folders

disk #2:      9Gb UW SCSI, 10K RPM, SCSI controller built in  
V: 8.7GB Jim's data? Largely free

#### Devices:

PCI: Matrox Meteor II frame grabber  
VCR: Panasonic PV-8450, 4-head HiFi  
Video Monitor: RCA XL100 13" color television

#### Licensed Software:

Visual Studio 6.0  
Office 2000  
Matrox MIL-Lite: Matrox imaging library  
Matrox Inspector  
Matrox Intellicam

#### Freeware:

EmTex  
Adobe Acrobat Reader  
GSView  
Winzip

#### System:

service tag: EF537  
express service code: 242-508-67  
FAT to NTFS conversion done

### 1.3.3 cyyoung.ee.psu.edu

130.203.198.166      residence of source code

#### Hardware:

Dell P6350 OptiPlex GX1/MT

Windows 2000

3 years service from Dell

350 Mhz Pentium II

100 Mhz bus

21" monitor

384 Mb ECC SDRAM, consisting of 3 128 Mb DIMMs

PCI vide card: 3dfx Voodoo5 5500 PCI, 64 MB SDRAM

*Old* PCI video card: ATI Rage Pro Turbo, 2X AGP with 8Mb SGRAM

PCI slot 1: SCSI controller

14-32X CD ROM, IDE

100 Mb Zip drive

integrated 3Com Fast EtherLink XL 10/100 ethernet card

floppy drive

built-in Sound Blaster compatible audio

mouse, keyboard

disk #1: 9.1 Gb UW SCSI, 7200 RPM

C: 2.0GB

F: 3.1GB, freeware, software

Y: 1.3GB, EMPTY

Z: 2.0GB, user folders

disk #2: 9.1 Gb UW SCSI, 7200 RPM

S: 4.9GB, source area

X: 3.6GB, EMPTY

#### Licensed Software:

Visual Studio 6.0 – from PSU EE

Office 2000 – from PSU EE

#### Freeware:

EmTex

Adobe Acrobat Reader

GSView

Winzip

#### Other:

service tag: EDFKM

express service code: 241-413-34

### 1.3.4 hornsby.ee.psu.edu

130.203.198.138

Hardware:

Dell P6350 OptiPlex GX1/MT  
Windows 2000  
3 years service from Dell  
350 Mhz Pentium II  
100 Mhz bus  
21" monitor  
384 Mb ECC SDRAM, consisting of 3 128 Mb DIMMs  
PCI video card: 3dfx Voodoo5 5500 PCI, 64MB SDRAM  
*Old* PCI video card: ATI Rage Pro Turbo, 2X AGP with 8Mb SGRAM  
PCI slot 1: SCSI controller  
14-32X CD ROM, IDE  
100 Mb Zip drive  
integrated 3Com Fast EtherLink XL 10/100 ethernet card  
floppy drive  
built-in Sound Blaster compatible audio  
mouse, keyboard

disk #1: 9.1 Gb UW SCSI, 7200 RPM

C: 2.0GB

G: 0.6GB, EMPTY

V: 3.8GB, EMPTY

Z: 2.0GB, user folders

disk #2: 9.1 Gb UW SCSI, 7200 RPM

F: 4.2GB, freeware and licensed packages

U: 4.2GB, EMPTY (but has install files)

Licensed Software:

Visual Studio 6.0 – from PSU EE

Office 2000 – from PSU EE

Adobe Illustrator

Adobe Framemaker

OpenNT

Freeware:

EmTex

Adobe Acrobat Reader

GSView

Winzip

Other:

service tag: EDFKP

express service code: 241-413-37

HP Scanjet Scanner attached

### 1.3.5 koufax.ee.psu.edu

130.203.198.238 In WEH's office

Hardware:

Dell P6450 GX1p/T+  
Windows 2000  
3 years service from Dell  
450 Mhz Pentium II  
100 Mhz bus  
19" monitor  
256 Mb ECC SDRAM  
8 Mb VRAM, 3D video card  
17/40X IDE CD ROM  
integrated 3Com Fast Etherlink XL 10/100 ethernet card  
floppy drive  
built-in Sound Blaster compatible audio  
Harmon Kardon HK195 speakers  
mouse, keyboard

disk #1:

14.4 GB EIDE  
C: 2.0GB  
F: 3.9GB, license, freeware, tex  
G: 3.9GB, weh's work  
H: 3.9GB, EMPTY

Licensed Software:

Visual Studio 6.0 – from PSU EE  
Office 2000 – from PSU EE

Freeware:

EmTex  
Adobe Acrobat Reader  
GSView  
Winzip

Other:

service tag: HOTJL  
express service code: 285-917-61  
Bios v 2.01.05 (?)

### 1.3.6 williemays.ee.psu.edu

130.203.198.237

Hardware:

Dell P6450 GX1p/T+  
Windows 2000  
3 years service from Dell  
450 Mhz Pentium II  
100 Mhz bus  
19" monitor  
384 Mb ECC SDRAM  
8 Mb VRAM, 3D video card  
17/40X IDE CD ROM  
integrated 3Com Fast Etherlink XL 10/100 ethernet card  
≥ 33.6K modem  
floppy drive  
built-in Sound Blaster compatible audio  
Harmon Kardon HK195 speakers  
mouse, keyboard

disk #1:

14.4 GB EIDE  
C: 2.0GB  
D: 3.9GB, Open?  
E: 3.9GB, freeware  
Z: 3.9GB, user folders

Licensed Software:

Visual Studio 6.0 – from PSU EE  
Office 2000 – from PSU EE

Freeware:

EmTex  
Adobe Acrobat Reader  
GSView  
Winzip

Other:

service tag: GYDDM  
express service code: 284-774-98  
NTFS  
Bios v 2.01.05

### **1.3.7 musial.ee.psu.edu**

130.203.198.167      Lexmark 1650n network laser printer  
1200 dpi  
4Mb memory  
Hardware address: 00200014AZE5  
Software: MarkVision Printer Utility.

## **1.4 Other Hardware**

1. HP Photo Smart printer – color photo printer

\*\* Very expensive to use.

\*\* Only to be used by permission of WEH!

2. HP Scanner

3. Olympus bronchoscope suite

Jim – write some material here:

1. Parts of olympus scope set-up
2. calibration program
3. tracking devices, David, phantom

## **1.5 Sun Configuration**

The Sun workstations in the lab have the following names: aaron, doubleday, gehrig. The Sun yastrzemski is in WEH's office. Sun Workstation specs are given below.

### 1.5.1 aaron.ee.psu.edu

Enterprise 3000 server

Main lab server, has home directories

Has installed packages (/usr/local) and lab software (/psu)

512 Mbytes RAM

Creator3D frame buffer – 24-bit color

three 4.3 Gb internal disks

629Mb swapspace residing on /tmp (internal disk #1)

OS: Solaris 2.7

4.3 Gb internal disk #1: home directories, swap, /opt, OS installation

    /export/home

    /tmp (swap space)

4.3 Gb internal disk #2: lab software, executables, manuals, demos

    /aaron2; also known as /psu

    Other systems can refer to /psu as /opt/psu – but not recommended

### 1.5.2 doubleday.ee.psu.edu

sparcstation20 128 Mbytes RAM

2 processors – 2 mbus cards

Sun SX frame buffer – 24-bit true color

1 Gbyte internal disk

4.3 Gbyte external disk

360 Mbyte swapspace residing on /dev/dsk/c0t3d0s1 (internal disk)

8mm tape drive

OS: Solaris 2.7

External disk:

    /dev/dsk/c0t1d0s6 – /opt2

    /dev/dsk/c0t1d0s7 – /work5

Sun 8mm tape drive

ADS hard disk, 4 Gbytes formatted;

    file systems: /opt2, /work5, /work6

Sun 1 Gbyte hard disk

    attached to doubleday?

    file systems: /opt?

hard disk, 2.1Gbytes formatted;

    attached to doubleday?

### 1.5.3 **gehrig.ee.psu.edu**

Ultra5 model 360  
360 MHz UltraSPARC-IIi processor  
128 MB RAM  
DRAM, 256 KB L2 cache  
8 GB internal disk  
1.44 MB floppy drive  
CD-ROM  
PGX24 graphics  
17" color monitor  
OS: Solaris 2.7

### 1.5.4 **yastrzemski.ee.psu.edu**

sparcstation10 in WEH's office  
64 Mbytes RAM  
GX frame buffer – 8-bit color  
sbus card used for sparcprinter  
sparcprinter  
1 Gbyte internal disk  
300Mbyte swapspace residing on /dev/dsk/c0t3d0s1 (internal disk)  
OS: Solaris 2.5

Sparcprinter **yaz**

**yaz** printer software in /opt/NeWSprint



## 2 Lab Administration

### 2.1 PC File Systems and Security

- Home directories are on the Z: drive (Z:\users\) of all PCs.
- Drives from S: to Y: are reserved for project source/exec areas on all systems. This convention and the Z: convention facilitate portability across systems for the Visual Studio packages (it needs absolute path names).
- E:\freeware – gscript, gsview, winzip on Clemente
- E:\license – Matrox Imaging Products on Clemente
- F:\freeware – gscript, gsview, winzip on CyYoung and Hornsby
- F:\license – Adobe Framemaker, Illustrator, typemaker on Hornsby
- F:\license – MS Office, Microsoft Developer Studio and OpenNT on all PCs (E: drive on Clemente).
- Security: Please create files only in your home directory or in C:\temp.
- Partition conventions for the PCs:
  - E: on CyYoung and Hornsby and F: on Clemente reserved for freeware and licensed software.
  - S: – Y: on all systems reserved for project source code, include files, and executables (software development partition).
  - Z: on all PCs reserved for home directories.
- Users have no access to unused partitions at this time.
- NO NEW PACKAGES ADDED WITHOUT WEH'S PERMISSION!

### 2.2 Image Databases

#### 2.2.1 PC Database

An image database exists on the D: drive of the PC paige.ee.psu.edu. It includes many CT case studies, old quicksee case studies, and other data.

The W: drive of paige has many micro-CT images.

The V: drive of clemente has much data from Jim Helferty, particularly pig data and airway avi files.

### 2.2.2 Sun Database

A variety of images exists in the directory `/aaron2/database`.

In this directory are various 3D images and images used in EE/CSE485/485.

Also, `/aaron4/database` serves as additional image database. MicroCT and other images are stored here.

A demo area exists in `/psu/demos`

The README file discusses the various demo areas. Subdirectories here often have extensive README or DOCUMENTATION files that describe how to run the demos.

## 2.3 Lab Web Page

The lab has a home page which can be accessed using any of the available world-wide web browsers (e.g., *netscape*). The address for the lab home page is

**`http://gehrig` or `http://gehrig.ee.psu.edu`.**

All user files associated with the lab home page reside on gehrig in `/opt/www/httpd/pub`.

*Note that `cobb.ee.psu.edu` also points to the lab webpage!* Until 1999, this was the computer that stored our lab's web page. Our web server runs a version of the popular Apache package. (It was NCSA previously.)

### 2.3.1 Personal Home Page on Lab Web Page

*You may have **your very own place** to store the web stuff.* By taking user (Shu-Yen) wan as the example, he made a `www/` directory under his home directory and changed the access mode as public readable and executable:

```
% cd /home/wan
% mkdir www
% chmod 755 www
```

He then put all my web stuff in `/home/wan/www` and change the access mode too:

```
% cd /home/wan/www
% chmod 755 *
```

When you type `http://gehrig.ee.psu.edu/~wan`, you will try to search, in order, the following files as his home page:

```
index.html
index.htm
home.html
html.htm
```

Please contact the webperson when you have made your page.

### 2.3.2 Home Page Maintenance

The assigned webperson maintains the lab home page. Changes to files for our page can be made in the directory:

```
gehrig:/opt/www/apache/htdocs
```

This directory includes everything.

## 2.4 PC Virus Scanning

For the PCs, CEDCC maintains Dr. Solomon's virus scanning. Joe Lanager of CEDCC has an administrator-level account on our machines, and he provides periodic automatic updates for new virus strains.

27 MAY 1998: WHEN WE GET NT SERVER RUNNING, WE SHOULD SET UP A DOMAIN AND ADMIT JOE LANAGER TO THIS DOMAIN. HE WILL THEN NO LONGER NEED AN ADMINSTRATOR-LEVEL ACCOUNT ON OUR SYSTEMS.

## 2.5 Lab ftp Area

The anonymous user can access our ftp site by usual ftp command or from a web browser (`ftp://aaron.ee.psu.edu`). To do this,

1. Type **ftp aaron.ee.psu.edu** at the prompt.
2. Type **ftp** at the login prompt.

3. Type in your complete email address at the password prompt.
4. Type **% cd pub/incoming/username** where “username” is the user’s ftp site you are accessing.
5. Get the desired files.

The internal directory for the ftp area is `/export/home/ftp`. Users can put items in their respective ftp directories (`/export/home/ftp/pub/incoming/username`). Outside people can ftp things to the lab by doing a **% cd pub/incoming** and placing the items in this directory.

## 2.6 Sun File Management and Home Directory Quotas

Users have limits on their home directory size. As general policy:

1. Images and large figures (.ps files, etc.) should not be kept in a home directory.
2. Large executables should not be a home directory.
3. Images, executables, large image-figures, tar files, and other large items should be placed in a data directory.
4. Users should regularly monitor and delete core files, xemacs files, latex extraneous files, \*.o files, and other unneeded files.

The following is a list of details on the lab’s home directory quota system.

1. Soft and Hard Limits. The soft limit is 100 MB, while the hard limit 110 MB. If someone exceeds the disk usage (under `/export/home`) of 100 MB, a warning message will show up next time he/she logs in. However, he/she is still be able to take more disk space until reaching 110 MB.
2. Number of files limit. Currently, no limit.
3. Time limit. The time limit for being over the soft limit is 1 week. If one should take more than 100 MB, he/she should reduce the usage within 1 week.
4. WEH has NO quota.
5. Get information about quota settings:

```

* Become as root
* repquota -va          => quota report for all
  quota -v <user name> => quota setting for <user name>
  quota -va             => quota report for all, with more detailed
                        info

```

## 2.7 Miscellaneous Sun System/Lab details

1. *The syslog user* is a special user whose home is **/home/syslog**. In this directory the file *syslog* contains logs of all mail sent to user syslog. Whenever important system news is sent to lab members, users should CC syslog to record a copy of the information.
2. *General System help* for machine system numbers, other past diagnostic information, etc., exists as text files in **/opt/psu/sys\_work**
3. Old user accounts are placed in a “graveyard.” The graveyard directory resides on aaron in **/export/home/graveyard**. Graveyarded accounts have no userid and are often compressed via gzip.
4. To increase swapspace on a Sun system, do the following:
  - (a) `% mkfile -v SIZEm pathname//` where SIZE is a number (so SIZEm is something like 180m for 180 megabytes) and pathname is an absolute pathname for the swapfile.
  - (b) `% swap -a pathname`

## 2.8 Procedures for Setting up New Sun Accounts

**Necessary Files to edit or add:**

globalrc, .cshrc, .login, /etc/passwd, /etc/.shadow, /etc/auto\_home

**Steps :**

1. **su** to root.
2. Type **admintool** to invoke the system administration tool.
3. Select **User Account Manager** within the tool.

4. Add an account for the new user using the **Add User** option under **Edit**. Any of the existing user accounts can be used for guidance. While setting up the new account using `admintool`:
  - (a) Select **None** for Naming service. We are not using NIS currently. Hence the account has to be set up on each of the individual machines.
  - (b) Be sure to select a **NEW** User ID #, different from existing User Id #'s. The convention followed is to select a User ID # one higher than the current highest User ID #. The same user ID # has to be selected on ALL machines. A good place to look for available user ID numbers is `/etc/passwd`. The third field in this file is the `userid`. Make sure that the number you choose (which should be the next sequential one available) is not already in use.
5. Set the items in the “Add User” window as follows:
  - (a) **User Name:** The person’s user name.
  - (b) **User ID:** The first available user id number.
  - (c) **Primary Group:** Generally, set this to “20” for “user.”
  - (d) **Comment:** The user’s real name.
  - (e) **Path:** `/export/home/yaz/username`.
  - (f) **Server:** `gehrig`
  - (g) **AutoHome Setup:** Check yes.
  - (h) **Permissions:** use default.
6. Copy the skeleton `.cshrc` and `.login` files from `/etc/skel` on `gehrig` to the home directory of the new user. Check to see if the permissions and ownerships of the files have been set correctly.
7. Edit the `/etc/auto_home` file on each machine to include the new user.

## 3 Equipment Maintenance and EE Staff

### 3.1 PCs

All PCs have full hardware and software maintenance support. Contract information appears below.

Dell's phone number is 1 800-626-8286.

As of July 1999, we have three-year hardware support and lifetime phone support for all PC computers. All contracts are operational (including paige). We have lifetime phone support and three year On-site parts and service support.

The contract dates and services tags are:

Clemente	Dell Precision 410	EFS37	Onsite Service till: 5/11/2001
Paige	Dell Precision 410	FMFRM	Onsite Service till: 7/29/2001
CyYoung	Dell Optiplex GX1	EDFKP	Onsite Service till: 5/5/2001
Hornsby	Dell Optiplex GX1	EDFKM	Onsite Service till: 5/5/2001
Willy Mays	Dell Optiplex GX1p	GYDDM	Onsite Service till: 10/20/2001
Koufax	Dell Optiplex GX1p	H0TJL	Onsite Service till: 10/27/2001

(H0TJL - thats a zero in there)

### 3.2 Sun Maintenance

Currently (October 2000), aaron is the only Sun under maintenance support.

Our contract number with Sun is NK20007235.

You can call 1-800-SUN-4USA for service calls.

The Sun gehrig has a one-year warrantee expiring on July 2001 (?).

### **3.3 EE Department Staff**

Bob Selfridge and Marsha Church are available as EE Computer Staff consultants for Sun and PC problems. Their office is on the 2nd Floor EE West, around the corner from the lab by the 2nd Floor EE West Men's room.

Bob Selfridge – [bselfridge@psu.edu](mailto:bselfridge@psu.edu)

Marsha Church – [msc2@psu.edu](mailto:msc2@psu.edu)



## **4 Miscellaneous Lab Items**

### **4.1 Shooting Photographs and Slides**

Use ASA100 speed film. Ektachrome is very good.

Shoot photos using the bulb (hand-controlled) speed setting. The recommended f-stop settings are f11 or f16. Keep the shutter open 8 seconds when shooting. WEH has used the Sun clock tool for timing. It is good to take multiple exposures. Have the room completely dark. Use a tripod and shutter release. The shutter release is good for control for the bulb setting. Also, try to get as close to the subject as possible. Fill the frame.

The recommendations above work very well for both 35mm slides and for color photographs.

### **4.2 Powerpoint to 35mm Slides**

Powerpoint slides can be converted into 35mm slides. Call 863-1026, Room 110 Business Admin 2 does this. As of 7/99, you can talk to Steve, who's email address is [sxb10@psu.edu](mailto:sxb10@psu.edu).

Before sending files to them for processing, you must change the page setup in Powerpoint to 35mm first. Steve needs the name, budget, fund, and telephone for processing.

*An expensive service!*

### **4.3 Phone Directory**

Many phone numbers are posted in the metal cabinets near the phone. The phone list for Eric Hoffman's lab is posted (Physiologic Imaging Directory).

Perfect Order (Sun Microsystems) 1-800-851-1281, Mechanicsburg, PA.

Ken Wilcox of Sun: 412-825-8600.

### **4.4 General BIBTEX area**

This BIBTEX area can be found under `/psu/BIBTEX` on the Suns. It contains many bibtex files, with lots of bibliographic entries. The chance is good that much of what you need is here. If you look at this area, you'll see files called `swift.bib` and `wan.bib`, among others; these files signify unique bibtex generated by past (and current!) people in our group.

Freely use this area. Updates and error fixes are also appreciated.

## **4.5 Lab Security**

The lab door should stay locked at all times. *No one is authorized to give others the combination the the Lab Door!*

Lab books and equipment should stay in the lab.

## **4.6 Long-Distance Phone Calls**

Long distance calls are not permitted from the lab phone. Personal long-distance calls can only be made with a personal calling card. Long distance calls to our collaborators and vendors are permitted with WEH's permission.

## 5 Code Specification

This sections describes the standard we use in the lab for software construction, documentation, and maintenance.

The files iMedian.h and iMedian.cpp illustrate many of the principles mention here. The appear in the directory /psu/manual/lab.

### 5.1 Variable Names and Declarations

1. Declare one (1) variable per line. This facilitates adding/removing variables during coding. The only exception to this rule is for generic counter variables i,j, ....
2. Use extra spaces between variable types, variable names, and comments to increase readability, for similar blocks of code. This is especially good to do for a series of variable declarations and blocks of assignment statements.
3. Put spaces around all variables surrounded by parenthesis "( )".
4. Use Hungarian Notation (HN) for variable names:

int	start with "n"	nMonthsInYear
char	start with "ch"	chMenuSelection
float	start with "fl"	flCheckBookBalance
BOOL	start with "b"	bFinished
double	start with "d"	dFoo
long int	start with "li"	liGrainsOfSand
short int	start with "si"	siMJordanHair
arrays	start with "ar"	arListofImages

This is expanded on below a bit to distinguish member variables and pointer variables.

5. The beginning prefix of the variable name starts with a lowercase letter. Additional prefixes making up the variable name should use uppercase letters.
6. Keep variable names under 20 characters or so if feasible; e.g.,

```
getValueOfIntegerParameterFromMenu
```

simplifies to

```
nGetIntegerParameter
```

Examples of (1. - 6.) appear below:

```
int      nEmployees;          // Number of employees
char     chMiddleInitial;     // A middle initial
float    flBankBalance;       // checkbook bank balance
char     chMenuSelection;     // character string from a menu item
BOOL     bFinished;           // Are we finished?
long int liGrainsOfSand;      // A lot of sand particles to count!
short int siMJHairs;          // Not much hair on M Jordan's head....
int      arListOfImages[5];   // An array

GetAge( int nYearBorn );
```

7. Group globally used variables (relative to a \*.cpp file) at the top of the file. That is, put all "int" variables in one contiguous block, etc.

Local variables can be declared at the spot they're first used; e.g., the local loop counter "t" is declared when it is first used

```
for( int t=0; t < m_nXdimension; t++ )
```

8. Capitalize the names of classes. For a given project, try to use the same small prefix for all classes; examples:
  - a. All MFC classes start with "C"
  - b. OpenGL classes all start with "Ogl"
  - c. VFX classes all start with "vc" (violates Greg's own capital rule!)

9. Do not use "register" variables. Compilers now handle this detail.
10. Use \*.h files to give a source (\*.cpp) file the necessary outside information.
11. Use the AFX.... concept for exporting class details....
12. Distinguish a class's member variables with prefix "m\_" and follow the Hungarian notation above. For example, m\_nEmployees could be an integer member variable for a class, whereas nEmployees could only be a variable internal to some function and not accessible in anyway.
13. Variables that are pointers can have a "p" for each pointer depth. For example, a member variable of type char\*\* might be called m\_ppchBitCount.
14. Do not use global variables. Variables should always be a member of some defined class.
15. PROPOSAL:
  - a. start the names of all of our classes with "M"
  - b. various projects can add to this prefix: "M3D" for the 3DNavigator

#### References:

- M. Williams, Essential Visual C++, Sams Publishing, pp. 42-43
- Greg Simon, M.S. Thesis, Appendix B, 1997.
- J. Lakos, Large-Scale C++ Software Design, Addison-Wesley, 1996. – USED THROUGHOUT THIS SPEC.

## 5.2 Source Code Files (\*.cpp and \*.h)

1. HEADER OF SOURCE FILE:
  - should give comments defining what all variables are. See attached example for iAssign.cpp. Reference:
  - S. Oualline, Practical C++ Programming, O'Reilly, 1997, pp. 39-40
2. Give comments for each logical block of code!!
3. Keep all source lines  $\leq 80$  characters. This helps in printing out copies on a standard-size printer.
4. Use carriage returns at the end of code lines. DO NOT USE TABS.

5. Use four (4) spaces for each level of nesting in a file.
6. Put blank lines between separate self-contained blocks of code.
7. After each CONTROL STATEMENT ( if , else , for , switch , case , while , do ), indent the subsequent block of code and set off by braces if the block has more than one statement.
8. Keep source files small, preferable under 5 pages.
9. Have separate files for separate groups of functions.
10. Throw exceptions and carefully account for error conditions.
11. Use destructors to delete memory no longer needed, especially for objects that use images.
12. Use a “const” definition to declare constants. Don’t use “define” statements.
13. An “inline” function definition is much better (instead of # define statements) , as this facilitates debugging. See Greg Simon’s thesis, pg. 104.
14. Include only necessary “include” (\*.h) files in a source (\*.cpp) file.
15. Include Gates:

One issue with breaking up code into small files too much is that include files might get included “too much” during compilation. Greg Simon mentions the idea of “include gates” on page 105 of his thesis for header files. Putting the following code in a source file:

```
#ifndef M3DinterfaceINCLUDED
#define M3DinterfaceINCLUDED
#include "M3Dinterface.h"
.
.
.... code for include file
.
.
#endif
```

will cause the include file "M3Dinterface.h" to be included only if it hasn't already been done somewhere else. If "M3Dinterface.h" had already been included during build by an earlier source file, then "M3DinterfaceINCLUDED" will have already been defined.

16. Name source files after the functions they contain.
17. Build projects with a sensible directory structure. Use directories to
  - a. break up large logical units
  - b. set up libraries (areas of related functionality)
  - c. plug-ins
  - d. internal executable code separated from GUI code
  - e. GUI code separated from executable code

### 5.3 Classes, Object-Oriented Programming

1. For classes, ALWAYS DEFINE CONSTRUCTORS AND DESTRUCTORS, even if the destructor isn't really important.
2. For classes containing IMAGE (OR SCRATCH) DATA, be sure to call destructors when the image data is no longer needed.
3. All functions must be prototyped in header files and defined in \*.cpp files. This should be done even for very simple functions.

4. The function of a \*.h file is to give users a set of "hooks" into the object and define implementation details.

Knowledge of internal workings/implementation of the object, as defined in associated \*.cpp files, should in no way be necessary to use the object.

5. In Class header files, list public members and methods first. Then, list protected/private members and methods.
6. \*.h files shall include:

necessary include files  
necessary library files

full usage instructions  
complete description of input and output parameters  
helpful comments  
examples

With this set-up, the user need only see the include file for an object to be able to use it. The user can then use the object as a "black box," given knowledge of the include file.

It is fine to have redundant comments about usage, etc., in \*.h and \*.cpp files.

7. FUNCTION PROTOTYPING: Do as follows (1 parameter per line). This eases subsequent changes, such as adding variables. Example:

```
int nMonthsInDayCount(  
    int    nDayCount,    // Number of days counted  
    float  flSeconds,    // Number of seconds counted  
    double dYears )      // A huge count for the total number of years
```

8. Declare Accessor member functions as "const" to clearly show the function's inability to modify an object. For example, in iAssign.cpp,

```
char * iAssign :: getName( ) const
```

9. When prototyping a function, declare a variable as "const" if the variable is not modified by the function.
10. Pass objects by reference, not by value. This is analogous to passing a pointer to a variable rather than the variable itself.
11. PROPOSAL:  
MAINTAIN A GENERAL CLASS LIBRARY (or libraries) FOR LAB USE.

Document the libraries on-line in html— easy, just convert abbreviated headers into html and give info on where the code is.



## 5.4 Source Modification and Maintenance

1. Revisions/improvements to existing code: Only changes to internal \*.cpp code is permitted. The \*.h interface will retain its past definition (i.e., function prototypes will not change). This guarantees existing usage dependencies will still work and will not require change.
2. Revision details must be recorded in the "Revisions" section of the header comments. Be sure to give the following information for a revision:
  - (a) When was revision done
  - (b) Who did the revision
  - (c) What was the revision made
  - (d) Version number of file that was revised.

When revisions are made in the code, do the following:

- (a) Add a comment explaining the revision.
- (b) Comment out the old section of code being revised, but DON'T DELETE IT.
- (c) Add a comment line:

```
/***** Revision Date:  MM/DD/YY  By: Jane Doe *****/
```

- (d) Place in new code.

3. Old versions of files should be saved.

- (a) On the Suns:

An old source file called "source1.cpp" should be called source1\_MMDDYY.cpp, where MM/DD/YY is the date of the revision.

The revised file is called "source1.cpp"

Same convention for \*.h files.

- (b) On the PCs:

We should start to use Visual SourceSafe, a version control system within the Visual Studio.

4. One person will be responsible for managing the changes that occur to a project. This person, the project master, will:
  - (a) monitor new code that is added and check to see that it is commented sufficiently, etc.
  - (b) authorize checking out old files that need revisions
  - (c) Check in completed revisions, per procedures (2.-3.) above
  - (d) do master rebuilding of executables
  - (e) maintain a beta version and a golden version of execs

## 6 Available Software Packages

Below is a summary of the available software packages. Subsequent pages describe what some of them do, how to use them, and the directories in which the packages reside.

### 6.1 Editing FILES, Creating DOCUMENTS, Drawing FIGURES

#### 6.1.1 PC

Name	Description
EmTexgi	PC-based Latex document builder
Word	sophisticated Text editor
Excel	spread sheet software
Powerpoint	presentation software
Access	database software
FrontPage	Web publisher
Outlook	email software
Image Composer	software for processing images
Adobe Illustrator	software for drawing figures
Adobe FrameMaker	typesetting software
Adobe Acrobat Reader	PDF browser
Adobe Premiere	sophisticated video editor
Snag-It	Image capture, edit

### 6.1.2 UNIX

All instructions for Sun-based running programs assume that your `.cshrc` file sources the "globalrc" file.

Name	Description
xemacs	sophisticated Text editor
xman	nice program for getting on-line Solaris info
latex	Standard program for document formatting
pageview	POSTSCRIPT Viewer
ghostview	POSTSCRIPT Viewer
ispell	Spell checker for an ASCII file
xdvi	Program to display latex-formatted documents
dvips	Program to print out latex-formatted documents
detex	Filter to remove tex and Latex commands from a file
ieps	Converts image to encapsulated postscript
xmgr	Program for drawing figures
latex2html	latex to html converter

## 6.2 IMAGE PROCESSING and DISPLAY

### 6.2.1 PC

Name	Description
VFX	object-oriented Image analysis package
Virtual Navigator	Virtual Bronchoscopy package
3D-Design	general 3-D image viewer
nv	MS-DOS 3D image-analysis package
matrox inspector	Matrox's image-analysis package
contalgo	a system for 3D path analysis
Tree Analysis	3D path analysis tool using contalgo
IPLview	Image Viewer (DICOM or Analyze formats)
ROISurfaceWriter	Makes ROIs for Virt Nav rendering
VNavInstaller	Configure tools for running Virt Nav

### 6.2.2 UNIX

Name	Description
vida	3D image analysis and visualization package
impromptu	Image analysis package
vfx	object-oriented Image analysis package
test_prog	vfx-based standalone C++ program to test individual functions
super3Dmegadisplay	Package for display of 2D/3D images
d4, D4	Superduper version of Super3Dmegadisplay
interseg	Package for interactive 2D/3D/4D segmentation
quicksee	Digital endoscopic system
texan	Package for texture segmentation
3DPIPE	Parallel image processing package
roi	program similar to impromptu
ISE	Interactive Segmentation Engine
xv	Image viewer and editor
contalgo	a system for path analysis

## 6.3 PROGRAM DEVELOPMENT, COMPILATION, and DEBUGGING

### 6.3.1 PC

Name	Description
Visual C++	C/C++ Integrated Development Environment
Visual BASIC	BASIC IDE
Visual Foxpro	database software
Visual InterDev	Web publisher
Source Safe	version control software
Rational Rose	large-scale C++ software development system
vtk	Visualization Toolkit
STL	Standard Template Library
OpenGL	SGI graphics library

### 6.3.2 UNIX

Name	Description
Sun SPARCworks	Sun C/C++ Development environment
Standard Template Library (STL) (C++ library)	
OpenGL	OpenGL graphics library
MesaGL	OpenGL-compatible graphics package
vtk	Visualization Toolkit
vfc-1	Visualization Foundation Class library (Keswa)
vfc-2	VFX's class library (G. Simon)
devguide	Package for building GUI
gcc	GNU C and C++ compiler
gdb	GNU debugger
dbxtool	Sun debugger

## 6.4 Miscellaneous packages

### 6.4.1 PC

Name	Description
OpenNT	X-Windows on Windows NT
eXalt	X-terminal software



### 6.4.2 UNIX

Name	Description
samba	hook up a laptop/PC to our Sun environment
makehdr	Creates analyze headers for .img files
makeheader	Interactively ceates analyze headers for .img files
update_header	Updates analyse headers
mathematica	Package for mathematical computation
matlab	Mathematical package for signal & image processing
pvm	Parallel Virtual Machine package

## 6.5 Packages for EDITING FILES, CREATING FORMATTED DOCUMENTS, DRAWING FIGURES

**xman** – Nice program for viewing Solaris manual pages.

**NOTES:** 1. A good way to get at global info quickly on groups of Solaris functions is the following. After “xman” is invoked, select the “manual-page” option, then “sections” suboption.

**latex** – Standard program for document formatting.

**manual:** See readily available books on Tex and LaTeX.

**how to run:** At the prompt, type

```
latex filename
```

where “filename.tex”, a latex ascii file, should exist.

**other comments:** 1. Use in conjunction with xdvi, dvips, and xfig  
2. A complete latex example appears below.

In the directory  
/work/users/weh/PAPER, you can see (and copy) a  
directory called "draft." This contains the complete  
text, figures, etc., for the journal paper Krishnan Ramaswamy  
and I recently submitted on 3D Dynamic Rendering.

This example has it all: text, complex equations,  
tables, embedded drawn figures, embedded images, and  
bibliography. It is a useful jump off point (in addition  
to about the first 50 pages of the latex book) for latex.

We have complete theses on-line also you can copy and  
fill in.

To compile the document I mention above, you run the  
following series of commands at the UNIX prompt:

1. latex paper (to build the paper from the file called "paper.tex")
2. bibtex paper (to compile the bibliography)
3. latex paper (to update cross-references in bibliography)
4. latex paper (final cross-reference update)

The hard-coded directories in the file "paper.tex" will have to  
be changed for your local set-up, maybe. Also, the figures  
embedded in the paper, I think, have hard-coded directory names  
for finding the figures.

Most of the time while writing a document, you just do one "latex" command. You only need to do the four commands when you update or fix your bibliography.

Note that I have been building up a general all-purpose bibtex area of citations, as we put out a fair number of papers. You, and anyone else, can use this. Papers you see cited in our papers will most likely have their citations contained in this master BIBTEX area.

**ghostview** – POSTSCRIPT language previewer.

**how to run:** At the prompt, type

```
ghostview filename
```

where "filename.ps", a postscript file, should exist, or just type

```
ghostview
```

and load the file from within ghostview.

**xdvi** – Program to display/preview latex-formatted documents.

**how to run:** First, run latex on the desired document; this will generate, among other files, a file called "filename.dvi". Next, at the prompt, type

```
xdvi filename
```

The document then pop ups in a window on the screen. Self-explanatory controls exist to view various pages, etc.

**other comments:** 1. Use in conjunction with latex, dvips, and xmgr

**dvips** – Program to print out latex-formatted documents.

**how to run:** First, run latex on the desired document; this will generate, among other files, a file called "filename.dvi". At this point, it is best to preview the document using "xdvi." When you are satisfied with the document, you can then print it out by typing at the prompt

```
dvips [options] filename
```

**other comments:** 1. Use in conjunction with latex, xdvi, and xmgr

**detex** – Filter for removing Tex and LaTeX commands from a file.

**how to run:** At the prompt, type

`detex filename`

where “filename.tex”, a tex file, should exist.

**other comments:** 1. Use in conjunction with latex.

**ieps** – convert image to encapsulated postscript.

**how to run:** type

`ieps filename`

where “filename” is an image file.

**NOTES: Usage:**

`ieps [-s slicenum] [-o outfile] [-p pixels-per-inch] infile`

**Synopsis:**

Converts slice *< slicenum >* of input file *< infile >* to encapsulated postscript. Writes output to file *< outfile >*. Can read Analyze format hdr files to determine image size automatically. For other formats, size must be manually specified.

**Default parameters:**

slicenum = 1, outfile = *<infile>.eps*, pixels-per-inch = 35

**Examples:**

Convert slice 40 of r16a to EPS, store in ./figure.eps:

`ieps -s 40 -o figure.eps /work/database/heart_sequence/r16a`

Convert camera image to EPS, store in ./camera.eps:

`ieps /work/database/ee455/camera`

**Note:** Slice numbers start at 1 and increment upwards; i.e., for a 95x95x90 image, the valid slice numbers are 1 to 90.

**xmgr** – A program for drawing figures.

**how to run:** At the prompt, type

`xmgr`

The main xmgr window pops up.

**other comments:** 1. Can use in conjunction with latex, dvips, xdvi.

**latex2html** – Latex to html converter.

From swift@ruth.ee.psu.edu Fri May 8 10:30:34 1998

Subject: Re: latex to html?

The latex to html converter is called latex2html in /usr/local/bin.

It has lots of options and creates many small files - I'd recommend trying it first on a small document that you copy to a new directory. It is also a bit flaky - not everything gets converted properly.

## 6.6 Packages for IMAGE PROCESSING AND DISPLAY

Several packages exist on the Suns for 3D image analysis, display, and visualization. A brief description of a few of these packages is presented in this section. The lab web page and previous student theses describe other packages.

On-line help is available on `cyyoung:S\manuals` for the PC-based packages.

**vida** – A 3D image analysis and visualization package that runs on the Suns. It is geared especially for medical images.

**manual:** A hard copy exists in the lab in a thin blue binder. It is somewhat out of date. Also, a few papers exist that document vida's design and use.

**how to run:** At the prompt, type

```
vida_start&
```

**location of source and executable code:** Source code is not available.

Executable code for the Solaris version of VIDA (VIDA 1.5) is in

```
/opt/vida/bin
```

Executable code for the SunOS version of VIDA is in

```
/usr/local/vida/bin
```

on ruth.

**impromptu** – An IMage PROcessing Module for Prototyping, Testing, and Utilizing image analysis processes.

**manual:** See the on-line manuals in

```
/opt/psu/src/manual/impromptu [AND impromptu_func]
```

Also, a paper by Gopal Sundaramoorthy et al. is available.

**how to run:** `impromptu` can be run in the interactive (window) mode and in command-line mode. If at the prompt, you type

```
impromptu
```

(No options.) you automatically go into interactive mode. If you instead type

```
impromptu [options]
```

then you go in the batch mode.

**other comments:** 1. Use in conjunction with `interseg` and `vida`. It can be directly accessed from `vida`. An `interseg` process automatically calls `impromptu`.

**test\_prog** – standalone C++ test program for testing individual functions.

**manual:** See source code's documentation.

`/opt/bin/test_prog.cpp`

Also, a paper by Gopal Sundaramoorthy et al. is available.

**how to run:** Use the makefile "Makefile.testprog" in `/psu/bin` to build the program.

The program's source file, `/psu/bin/test_prog.cpp` offers many comments on how to add a function and change the test image.

You can use one of two types of images:

1. A hard-coded image that can be dumped onto the screen or saved.
2. Load (and save) a regular image (.img, .hdr).

**other comments:** 1. It is particularly useful for testing binary-valued ops.

2. It is very useful for seeing precise answers and known gray-level values. Thus, it is useful for small known ground-truth tests.

**location of source and executable code:** `/psu/bin/test_prog.cpp`

Makefile

`/psu/bin/Makefile.testprog`

**super3Dmegadisplay** – A simple, quick system for displaying 2D/3D images. You can leaf through slices, magnify images, stretch the gray-scale, see gray-scale values, etc.

**manual:** Type at the prompt

`sdisplay`

to get a summary of the command-line options.

**how to run:** Type at the prompt

`sdisplay [OR mdisplay] [options]`

**other comments:** 1. `sdisplay` is for the display of a single image.

2. `mdisplay` is bringing up a sequence of displays for a number of same-prefixed images; e.g., `mdisplay r1*` will run `sdisplay` for all images prefixed "r1."

**location of source and executable code:** Source is in

`/work/psu-sources/Super3DMegaDisplay`

Executable is in

`/work/psu-bin/sdisplay [OR mdisplay]`

**NOTES:**

1. The code was compiled in SunOS. The executable runs on all Sun systems.

**d4,D4,mD4** – A superduper version of `super3Dmegadisplay` developed by Tun-Feng Hung.

**manual:** An on-line manual is available in `/opt/psu/manual/d4`.

**how to run:** Type

`d4`

or

`D4`

at the prompt.

**other comments:** 1. Need to be in open windows.

**NOTES:** 1. `mD4`

is a SCRIPT version that permits multiple invocations of D4; it also allows command-line options. For example, the command

`mD4 -m2 -x r16*`

will pop up D4 for instances of images beginning with “r16”; all of these instances will have 2x magnification and be displayed with the sagittal orientation (per option ‘x’).

**interseg** – A system for interactive 2D/3D/4D image segmentation.

**manual:** In the directory

`/opt/psu/manual/interseg`

You can type

`xdvi thesis`

to get information on how to run interseg. Also, several papers exist that document interseg’s operation and design.

**how to run:** At the prompt, type

`interseg`

**other comments:** 1. Must be in open windows.

2. Used in conjunction with `impromptu`, to run the image-analysis processes that interseg constructs. Further, it can run with `vida`.

**quicksee** – A digital endoscopic tool by Krishnan Ramaswamy.

**how to run:** At the prompt, type

`vset TrueColor 24`

`quicksee`

**other comments:** 1. Must be in open windows.

2. This version uses multiple threads AND the Rasterflex software for 24-bit color displays and, hence, runs on gehrig only.

**NOTES:** 1. A serial version of the program which does not use threads is available in /work/psu-sources/visualize. The corresponding executable is in /work/psu-bin. This program uses 8-bit color for its displays.

**texan** – A package for segmentation and analysis of textured images, developed by Tom Weldon.

**how to run:** At the prompt, type

```
texan
```

**other comments:** 1. Must be in open windows.

**3DPIPE** – A parallel-processing 3D image analysis package developed by Rod Swift. **Old!!**

**manual:** See Rod Swift/Bill Higgins. The user's guide for this package is available in the file /cyyoung/swift/thesis/thesis.dvi (Appendix A).

**how to run:** At the prompt, type

```
3dpipe {image} {fx} {fy} {fz} {process filename}
```

where *fx*, *fy*, and *fz* are the dimensions of *image* in the *x*, *y* and *z* directions and *process filename* is the name of the process file that is to be run by 3DPIPE. The format for the image is raw 8-bit binary data. Each byte of the data thus represents a single voxel value that must range between 0 and 255. The name of the image file as it exists on disk must have the file extension *.img*. The user omits this suffix when running 3DPIPE. For example, `3dpipe r16a 90 90 95 process1` See the User's Manual above for additional details.

**other comments:** 1. Must be in open windows. The PVM daemon must be running.

**location of source and executable code:** Source is in

```
/home/swift/pvmwork/sources
```

Executable is in

```
/home/swift/pvm3/bin/SUN4SOL2/3dpipe
```

**roi** – A program, similar to *impromptu*, for prototyping image-analysis processes. ROI is being superseded by *impromptu*. **REALLY old!! From WEH's Mayo days [pre-1989].**

**available systems:** SunOS systems only (ruth).

**manual:** An on-line manual exists in

```
/opt/psu/src/manual/roi
```

Use *xdvi* to view the *\*.dvi* file there. Also, a paper manual exists on the bookshelf.

**how to run:** At the prompt, type



`roi`

This runs a shell script that calls either `roi_master` (non-openwin) or `x_roi_master` (openwin).

**other comments:**

1. Only runs on SunOS systems.
2. CANNOT be in open windows, unless `x_roi_master` is run.
3. Being superseded by `impromptu`, but `impromptu` does not yet have all of `roi`'s functions.

**location of source and executable code:** Source is in

`/work/psu-sources/roi/master`

Executable is in

`/work/psu-bin`

which has the `roi` script, `roi_master`, and `x_roi_master`.

**ISE** – The following separate functions make up this package: `convert`, `gpv`, `lpv`, `lpvres`, `mipv`, `rpvres`, `spv`, `update`, `ws_filter`, `ws_mipv`, `ws_update`. Programs by Mike Hansen for relaxation labeling and watershed analysis. This is the so-called "ISE" package (for Interactive Segmentation Engines).

**manual:** See Mike Hansen's manuals in one of two directories:

`/opt/psu/manual/ise[/ise-man OR /thesis-793]`

and type at the prompt

`xdvi iseman [OR thesis]`

**how to run:** Type the name of the appropriate command at the prompt. See the manual to know the correct order.

**other comments:**

1. Specialized versions of these functions are built into `impromptu`.

## 6.7 Packages for PROGRAM DEVELOPMENT, COMPILATION, and DEBUGGING

**sparcworks** – Sun developer’s environment including debugger, analyzer, compilers, and utilities.

**manual:** On line - type “man sparcworks”. Also, run the Answerbook from doubleday - type “answerbook” and select the appropriate topic.

**how to run:** Type

```
sparcworks
```

**other comments:** A copy of the license file is located on each machine in the file:

```
/opt/SUNWspro/license_dir/sunpro.lic,sp
```

**location of executable code:** /work5/SPARCworks/SUNWspro

**OpenGL** – Standard graphics library developed by SGI. QUICKSEE now uses it for the CubeTool and SurfaceTool. It runs well only on aaron. When an application calls OpenGL routines on a different system, then a switch is made to MesaGL. *Old, but used by quicksee4.0.*

**available systems:** aaron

**manual:** Some brief installation info is in the cabinet.

**how to run:** Not applicable. See quicksee for an example of incorporating OpenGL. Also, see the NOTE below on examples.

**location of library:** is in

```
/opt/SUNWsdk/sdk_2.5/GL
```

**NOTES:** 1. Nice examples of using OpenGL in systems is in

```
aaron:/opt/SUNWsdk/sdk_2.5/GL/contrib
```

**MesaGL** – Standard graphics library compatible with OpenGL. Much slower than OpenGL, but runs just like OpenGL. Must be used for now when an application calls OpenGL routines on a system other than aaron.

**how to run:** Not applicable. See quicksee for an example of incorporating MesaGL. Also, see the NOTE below on examples (from OpenGL).

**location of library:** is in

`/usr/local/lib/libMesaGL.so`

and several other libs/files in `/usr/local/lib`. Include files are in:

`/usr/local/include/GL`

**vtk** – The Visualization Toolkit, from Lorensen *et al.* Used heavily by the Virtual Navigator and 3D-Design. Also, used by QUICKSEE by the CubeTool and SurfaceTool. The Visualization Foundation Class (VFC) library of Mduduzi Keswa uses vtk.

**manual:** No on-line doc, but you can see the book “The Visualization Toolkit” by Lorensen *et al.*

**location of library:** is in

`/usr/local/lib/libvtk.so`

Include files are in:

`/usr/local/include/vtk`

**vfc - 1** – Visualization Foundation Class library. See Mduduzi Keswa’s thesis. Can be used to build stand-alone applications. Needs OpenGL and vtk.

**manual:** M. Keswa’s thesis.

**location of library:** is in `/psu/src/quicksee_4.0`

**vfc - 2** – Greg Simon’s class library for building vfx. The volume class and many others can be reused for 3D volume imaging applications.

**manual:** G. Simon’s thesis and user manual in

`/psu/manual/vfx`

**how to run:** need environment variable VFX set; done by the lab’s globalrc file. To run, just type “vfx.”

**location of library:** is in `/psu/src/vfx`

**devguide** – Package for building GUI’s in OpenWindows. *Devguide was used for building the front-end graphical user interfaces for all the SUN-based image processing packages developed in the lab. It is old, but we need to keep it around!*

**manual:** See Devguide manuals in the lab and also 'Answerbook' in Solaris.

**how to run:** At the prompt, type

```
devguide
```

**other comments:** 1. Must be in open windows.

**location of source and executable code: SunOS version:**

Source code is in

```
/usr/local/3.0_Devguide
```

Executable code is in

```
/usr/local/3.0_Devguide/bin
```

**Solaris version:**

Source code is in

```
/opt/SUNWguide
```

Executable code is in

```
/opt/SUNWguide/bin
```

1. The Solaris version is **Devguide 3.0.1**.

**gcc** – GNU C and C++ compiler

**manual:** At the prompt, type

```
man gcc
```

or

```
xinfo gcc
```

**how to run:** At the prompt, type

```
gcc [options] filename
```

or

```
g++ [options] filename
```

where filename, a 'C' or 'C++' file, should exist.

**other comments:**

1. There are 2 different versions of the gcc compiler installed currently; one for the SunOS systems and one for the Solaris systems.
2. Use in conjunction with gdb for debugging.

**location of source and executable code:** Source for the SunOS gcc compiler is not present (lack of space).

Source for the Solaris gcc version is archived in

`/opt2/ftp/gcc`

on doubleday.

Executable for the SunOS version is in

`/usr/local/opt/cygnus-sun4-2.0/bin`

Executable for the Solaris version is in

`/opt/gnu/bin`

on gehrig and exported to yaz, doubleday, and aaron.

**NOTES:** 1. The Solaris version of gcc is 2.5.8. *A bit old....*

2. The libraries and include files and info files for the Solaris gcc compiler are in `/opt/gnu`.

3. The SunOS version of gcc is cygnus-2.3.3.

4. The libraries and include files and info files for the SunOS gcc compiler are in `/usr/local/opt/cygnus-sun4-2.0`.

## 6.8 Miscellaneous Packages

For PC-Sun interactions, creating/manipulating image headers, and mathematical processing.

**samba** – Allow a laptop/PC user to access files in our Sun environment

**available systems:** aaron only

**how to run:** Run automatically on aaron at boot-up from within `/etc/rc2.d/S99samba.server`

**other comments:**

**location of configuration files:** `/opt/ATsamba`

Executable shellscript is on

`/etc/rc2.d/S99samba.server`

Executable binaries are on

`/opt/ATsamba/bin`

**NOTES:** 1. See the documentataion within the `/opt/ATsamba` directory on aaron.  
2. run `/opt/ATsamba/smbclient` to debug configuration issues

To access a Solaris disk from within Windows Explorer (Windows 95/NT):

1. go to Tools... Map Network Drive.
2. Select a drive letter on your PC that's not in use
3. type:  
`\\aaron\userid`  
where userid is your login id on aaron.

**makehdr** – Create Analyze headers for .img files on the Suns.

**manual:** Type 'makehdr' without any arguments to get information on usage.

**how to run:** Type

`makehdr filename [arguments]`

where filename.hdr is the file to be created.

**location of source and executable code:** Source code is in

`/opt/psu/sun4os/src`

Executable code is in

`/opt/psu/sun4os/bin`

**makeheader** – Interactively create Analyze headers for .img files

**how to run:** Type makeheader

**other comments:** 1. Lets the user define x,y, and z dimensions, delta x,y, and z pixel dimensions, 8 or 16 bit data type, scale Max and Min, and a comment for the image.

2. This is newer than makehdr.

**location of source and executable code:** Source code is in

`/opt/psu/src/Utilities`

Executable code is in

`/opt/psu/bin/`

**update\_header** – Updates Analyze headers so that the datatype flag is appropriately set. On the Suns.

**how to run:** Type

`update_header filename`

where filename.hdr, a header file, should exist.

**location of source and executable code:** Source code is in

`/home/jmr/update_header`

Executable code is in

`/work/psu-bin/`

**mathematica** – Interactive system for doing mathematical computation.

**available systems:** Sun systems only.

**how to run:** At the prompt, type

`math`

**pvm** – The parallel virtual machine. A data transfer protocol and software package for implementing 'multiprocessor' architectures by using multiple machines. *Pretty old, as the MasPar parallel-processing computer was sent to salvage as of 1998.*

**available systems:** All machines including cyyoung (this was an old dec computer!)

**manual:** See

`PVM Manual`

on the bookshelf

**how to run:** Type

pvm

at the prompt.

**other comments:** 1. Need not be in openwindows.

**location of source and executable code:** Source code is in

`/home/swift/pvm3/src`

on the SUN machines and in

`/user/users/swift/pvm3/src`

on cyyoung.

Executable code is in

`/home/swift/pvm3/lib`

on the SUN machines and in

`/user/users/swift/pvm3/lib`

on cyyoung.

- NOTES:**
1. Software is currently being developed to allow IMPROMPTU to use routines on the MASPAR via PVM. Also, IMPROMPTU batch files can be forked to multiple SUN machines, for improved performance.
  2. The overall purpose of PVM is to distribute a computational load across several machines. PVM automatically converts data types (i.e., SUN vs DEC formats) and has many powerful features to create a dynamic 'architecture' of processors.
  3. An example of how it is useful is as follows: A 'master' program can be written on one machine and 'slave' programs can be written on other machines. The master calls the slave programs, distributes information and then waits until the slaves have finished working. The slaves then send the results back to the master.



## 7 BACKUPS

### 7.1 PC Back-up Plan

The PC Backup Plan is designed to backup data on Paige, Clemente, CyYoung, and Hornsby. Paige is running Microsoft NT server and contains the 4 mm internal backup tape device. The plan includes full backups of all the computers every three months plus backups of development work three times a week.

#### 7.1.1 Hardware and drivers

The tape device is a Seagate 4mm internal tape drive. This device requires upgraded drivers for the tape driver and the AIC78xx SCSI controller. The tape driver is titled `stdat4`, and is installed from a floppy disk titled “Seagate SCSI DAT Tape Driver v1.00”. The SCSI driver is titled `aic78xx` and is installed from a floppy disk titled “7800 Family Manager Set Ver 3.01”. Both floppy disks and a manual for the tape drive are located in the Paige installation binder in the lab bookshelf.

#### 7.1.2 Inserting and Removing Tapes

The tape device is an internal tape drive located in the 5.25 inch bay on Paige, below the CD ROM drive. Tapes can be manually inserted and removed without the need for logging into the account. There are two small rectangular shaped buttons on the front of the device. Pressing and *holding* the button the right causes the orange light to flash, and after about twenty seconds, the tape will eject. Insert a tape into the device with the magnetic material towards the back of the computer and the pickup wheels on the tape facing downward. Push the tape into the device until the drive accepts it.

#### 7.1.3 Backup Software

The backup software used for the PC plan is a Seagate Software application titled “Backup Exec 7.0 for a Single NT Server”. This software will allow the backup of a single NT server plus an unlimited number of NT Workstations. The software resides on Paige and can be operated by a System Administrator or a Backup Operator. The Backup Exec software bundle including CD ROM and manual are located in the lab software bookshelf. Unfortunately, email notification of the backup status does not work since Microsoft Exchange Server is not installed.

## 7.2 PC Tape Rotation Schedule

As soon as the backup software was running a “Day One” backup was made of all the PC systems. This tape is not to be overwritten and will ensure a clean copy of the operating system to refer to if needed. The PC backup schedule is designed to be a Grandfather-Father-Son system. There will always be archive history that is three months old, three weeks old, and two days old. The Grandfather is a backup of the entire system every three months so that there will always be a backup history going back at least three months. The Father is a system of four tapes that backup the developement work every Friday. There should always be a Friday backup that is at least three months old. The son tapes are backups of the developement during the week that includes the Monday, Wednesday, and most recent Friday tape.

1. “Day One” Backup:

The purpose of the Day One backup is to have a clean operating system to refer to in case the other backup tapes are infected with a virus. This is a two tape archive that contains a full backup of all data on Paige, Clemente, CyYoung, and Horsby. These tapes are never to be overwritten.

2. Three Month Backup:

This is the Grandfather backup that consists of a complete backup of all data on Paige, Clemente, CyYoung, and Horsby. This backup requires two 12 Gig tapes per backup. On the next backup two other tapes are used so that there will be an archive at least three months old. Two tapes will be used for Winter-Summer and two tapes will be used for Fall-Spring, requiring a total of four tapes. This backup is not automated and will be done manually by the backup operator.

3. Routine Monday-Wednesday-Friday Backup:

These are the routine backups that include Father-Son backups of developement work and the system folder of all PC's and fits on a single 12 Gig tape. The backups are automated for late night every Monday, Wednesday, and Friday, so the operator only needs to insert to proper tape into the drive without needing to log into the computer. There are four tapes for Friday titled "Friday 1", "Friday 2,5", "Friday 3", and "Friday 4" corresponding to the occurance of that friday in the month. If there happens to be a fifth Friday in the month then "Friday 2,5" should be substituted. This will allow that there is always a three week history for the developement work. The Monday and Wednesday tapes are reused every week since these are only used for a two day backup.

The files included in the Monday-Wednesday-Friday Backup's are:

Paige::C:Drvlib

Paige::C:Winnt

Paige::Z:

Clemente::C:Drvlib

Clemente::C:Winnt

Clemente::Z:

CyYoung::C:Drvlib

CyYoung::C:Winnt

CyYoung::Z:

CyYoung::S

CyYoung::vfx

CyYoung::work

Hornsby::C:Drvlib

Hornsby::C:Winnt

Hornsby::Z:

### 7.3 Sun Back-up Plan

This is our Sun Back-up Plan. The 8mm Tape section below provides a few more details.

1. A COMPLETE SYSTEM BACK-UP of file systems on all Suns is done at the beginning of each month. This back-up requires FIVE Exabyte tapes. One is used on the 1st, another on the 4th, another on the 7th, another on the 10th, and the last on the 13th. The system administrator is sent a message each month as a reminder to do the back up. A report of each backup is also sent to the system administrator when a backup job is finished.
2. REGULAR THRICE-WEEKLY BACK-UPS of /export/home and /psu on aaron.
  - a. Each Sunday, Tuesday, and Thursday night, a reminder is sent to the back-up person.
  - b. The back-up person uses ONE Exabyte tape each Monday, Wednesday, and Friday.
  - c. The Friday tape is saved for the month as a weekly back-up.
  - d. If the back-up fails, the back-up person receives a message stating this.
  - e. The tapes from the previous month are recycled (reused) for the next month.

### 3. Questions:

- a. Where are cron jobs for these back ups and reminders?
- b. What exactly is backed up during the monthly backups?
- c. Other than the reports sent to the persons in charge of the backup stuff, where can I find more information of the backup job? Answers appear below.

3-a: The cron jobs for daily/monthly backup are set on doubleday. You may see the contents by becoming as root and typing "crontab -l".

3-b: Right now, our monthly backup is very comprehensive. It includes:

Tape 1: aaron:/export/home

Tape 2: aaron:/aaron4

Tape 3: aaron:/aaron2 => including /aaron2/psu, /aaron2/database, etc.

Tape 4: aaron:/aaron3/opt => /software

Tape 5:

gehrig:/opt\_psu/publisher

doubleday:/work5

doubleday:/work6

doubleday:/opt2

cobb:/opt/work7

cobb:/opt/www/httpd/pub

ruth:/var/spool/mail

ruth:/work

3-c: The backup scripts reside in /usr/local/backup. The monthly backup scripts are backup.export\_home, backup.aaron4, backup.aaron2, backup.aaron3\_opt, backup.others. The daily backup script is backup.daily. Each backup script stores the information during the course of backup in a subdirectory named from the extension file name of each script itself. For example, the backup information of backup.export\_home is stored in /usr/local/backup/export\_home. In the subdirectory, you may find an error report and the table of contents of the backup.

## 7.4 Old Back-up Plan – superseded 6/1/98

Each Monday Wednesday and Friday evening the following filesystems are backed up on the 150 Mbyte tape drive on cobb:

```
DIRS="/gexport/home/yaz ./dexport/home/doubleday"
```

where gexport/home/yaz is the temporarily mounted filesystem /export/home/yaz from gehrig, and ./dexport/home/doubleday is /export/home/doubleday on doubleday.

To avoid difficulties with automounter timeouts, each filesystem that is not local to cobb is mounted in a temporary directory /tmp\_mnt\_bak during the backup.

The script file used to do the backup is /usr/local/backup/backup.cobb, and it takes a single argument – the account name of the person who is to receive an e-mail message describing the status of the backup. The current crontab entry on cobb is:

```
15 1 * * 2,4,6 cd /usr/local/backup; /usr/local/backup/backup.cobb tpw
```

A sample mail message indicating the status of the backup is:

```
From root Sat Aug 20 03:10:39 1994 To: tpw Subject: backup cobb msg Content-Length: 171 X-Lines: 5
```

here is the backup status:

```
begin backup : Sat Aug 20 01:15:01 EDT 1994: mount completed: tar completed: umount completed: backup completed Sat Aug 20 03:10:27 EDT 1994
```

A sample mail message for an unsuccessful backup is:

```
From tpw Sun Aug 7 18:24:22 1994 To: tpw Subject: backup cobb msg Content-Length: 102 X-Lines: 5
```

here is the backup status:

```
begin backup : Sun Aug 7 18:24:02 EDT 1994: XXX=== TRAP ERROR ===XXX
```

When an error is found, detailed error messages can be found in /usr/local/backup/cobb/ERR. Only severe errors will result in the error message above, in general if the tar command encounters files where there are permissions problems, these will be recorded in ERR but not reflected in the mail message. So it is a good idea to occasionally check the ERR file. Generally these errors only occur with cyyoung filesystems since root on cobb is not equivalent to root on cyyoung.

Finally, a table of contents for the backed up filesystems is saved in /usr/local/backup/cobb/TOC.

## 7.5 Old Plan: ruth

Essentially the same procedure as cobb except that the backed up filesystems are:

`DIRS="/opt/psu /usr/spool/mail /usr/local/backup/backup ./cyyoung/usr/users /work/psu-sources"`

In this case only `cyyoung/usr/users` and `/opt/psu` need to be temporarily mounted (from `cyyoung` and `gehrig` respectively) since the other filesystems are local to `ruth` at present. Errors and tables of contents are in:

`/usr/local/backup/ruth`

and the crontab entry is:

`15 1 * * 2,4,6 cd /usr/local/backup ; /usr/local/backup/backup.ruth tpw`

See the foregoing description of `cobb` for more details.

## 7.6 8mm Tape Back-up

The 8mm tape drive on `doubleday` is used to do monthly backups of many of the more important filesystems. NOTE: YOU MUST USE EXABYTE DATA TAPES, NOT 8mm VIDEO TAPES. The mechanics of this backup are identical to `cobb`, non-local filesystems are temporarily mounted on `/tmp_mnt_bak` so the automounter does not unmount them during backup. The script for this backup is:

`45 2 15 * * /usr/local/backup/backup.dat.dd`

This backup is done on the evening of the 14th of every month, the filesystems presently backed up are:

`DIRS="/export/home/doubleday ./gexport/home ./work ./usr/local "`

`DIRS=" DIRS ./cyyoung/usr/users ./scratch ./gopt"`

where `./gopt` refers to `/opt` on `gehrig`.

## 7.7 special

Several special file areas outside the usual nitely backup filesystems are copied to these filesystems using crontab entries. Thus they are indirectly included in the backup. The relevant crontab entries are:

on all machines:

`15 0 * * 2,4,6 cd /usr/local/backup ; /usr/local/backup/tar.etc`

which backs up `/etc` from the machines into directory `/usr/local/backup/backup`

and on `doubleday`:

`10 0 * * 2,4,6 cp /usr/local/backup/backup.cobb /usr/local/backup/backup`

`11 0 * * 2,4,6 cp /usr/local/backup/backup.ruth /usr/local/backup/backup`

```
12 0 * * 2,4,6 cp /usr/local/backup/backup.dat.dd /usr/local/backup/backup
```

backs up the backup scripts.

## 7.8 Level 0 Dump on Solaris 2.X

These are the prerequisites for performing level 0 dump:

- Place the tape, 8mm in our case, in the drive that is attached to *doubleday*.
- Become as *root*
- Bring the system down to single user mode

Before bringing the system down, make sure there is no other users running their process on reside on some file systems by entering **w** on the command line to check who is on this machine, **ps -ef** to check what processes are running, or **fuser -c -u mount\_point** to check who is on certain *mount\_point*.

If killing processes is necessary, enter **kill -9 process\_id**, where *process\_id* is shown in the output of **ps -ef**, or enter **fuser -c -k mount\_point** to kill all the processes running on the file system.

In the following are the steps of level zero dump:

1. **shutdown -i0 -g120 -y**

This will bring the system to run level 0, in 120 seconds, without having to have further confirmation. The other users logging in this machine will be automatically notified.

2. **boot -s**

Enter single user mode. This is valid when there is an **ok** prompt. If your (older) system prompts you **>**, you should enter **b -s**.

3. Enter root password for system administration.

4. Enter **df -k** or check the contents of */etc/vfstab* to determine which file system you are going to dump, eg. */dev/dsk/c0t3d0s2* or */aaron4*.

5. **ufsdump 0cuf doubleday:/dev/rmt/0 /dev/dsk/c0t3d0s2** or **ufsdump 0cuf doubleday:/dev/rmt/0 /aaron4**

For details of the parameters, please refer to Part 9 of Solaris 2.X Administration Guide I.

6. `ufsrestore tf doubleday:/dev/rmt/0`

Browse the table of contents on the tape

7. Press Ctrl-D to bring the system back to run level 3, usual running mode.